

Материалы заданий Межрегиональной олимпиады школьников имени И.Я. Верченко по информатике и компьютерной безопасности

Содержание

ХIII Межрегиональная олимпиада школьников имени И.Я. Верченко по информатике и компьютерной безопасности (2018-2019 учебный год).....	2
9-10 КЛАССЫ	2
Условия задач отборочного этапа	2
Условия задач заключительного этапа	6
11 КЛАСС.....	14
Условия задач отборочного этапа	14
Условия задач заключительного этапа	18
Критерии оценивания заданий	26

**ХІІІ Межрегиональная олимпиада школьников имени
И.Я. Верченко по информатике и компьютерной безопасности
(2017-2018 учебный год)**

9-10 КЛАССЫ

Условия задач отборочного этапа

Задача 1

Начинающий программист решил свой пароль от аккаунта социальной сети защитить от просмотра третьими лицами.

Листинг.

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char s[100], p[100];
    strncpy(p, "*****", 100);
    printf("Enter password: ");
    scanf("%s", s);
    if (!strcmp(s, p, 100)) { printf("Password from account:
*****"); }
    else { printf("Wrong password!"); }
    return 0;
}
```

С этой целью он скомпилировал программу (см. *листинг*), заменив символы «*» на символы паролей: пароль от программы и пароль от аккаунта. В результате получилась программа, которая при запуске спрашивает пароль от программы, и если он введен успешно, то выводится пароль от аккаунта.

Через некоторое время оба пароля были забыты. Имея файл PROG.EXE, помогите программисту восстановить забытые пароли.

В ответе укажите оба пароля **через запятую без пробелов с учетом регистра** в формате:

ПарольОтПрограммы,ПарольОтАккаунта

Ответ: star2,aRmAdA

Задача 2

В исполняемый файл HELLO.EXE (см. приложение к задаче) было внедрено текстовое сообщение. При этом сам файл корректно выполняет все функции. Известно, что для того, чтобы отметить место внедрения сообщения, были

использованы 2 одинаковых однобайтовых метки, между которыми идёт сообщение:

Метка (1 байт)	Сообщение (X байт)	Метка (1 байт)
-------------------	-----------------------	-------------------

Какое сообщение было внедрено в исполняемый файл? В ответе укажите только само сообщение с учетом регистра и пробелов.

Ответ: Check security (метка 0xD7)

Задача 3

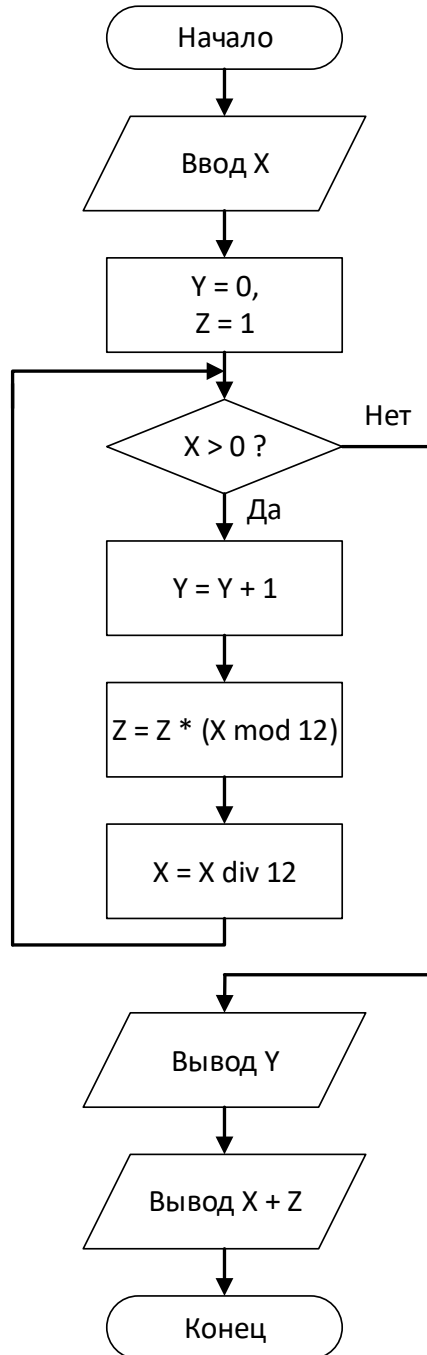
На рисунке представлена блок-схема алгоритма обработки целого числа X, вводимого пользователем.

Используемые в блок-схеме операции:

div – целая часть от деления;

mod – остаток от деления.

Какое минимальное число необходимо ввести, чтобы было выдано 3, а затем 48?



Ответ: 224

Задача 4

При перехвате трафика по сети нарушителю удалось получить хэш-образ пароля пользователя для входа в защищенную систему. Для того, чтобы узнать правильный пароль, он решил перебрать все возможные пароли, вычисляя их хэш-образы и сравнивая с известным. Сколько нарушителю потребуется времени (в секундах), если известно, что время проверки одного пароля – 10 секунд, пароль не может быть пустым и его максимальная длина составляет 9 символов, используется алфавит, состоящий из 8 символов.

Ответ: 1533916880

Задача 5

Локальная сеть расположена в одноэтажном здании, представляющем собой 2018 комнат, с коридором между ними, проходящим через всё здание и разделяющим четную и нечетную стороны нумерации. Во всех комнатах размещено по одному компьютеру. Каждый N -ый компьютер соединен отдельными кабелями с $(N-1)$ -ым и с $(N+1)$ -ым компьютерами, где $2 \leq N \leq 2017$. 1-ый и 2018-ый компьютеры также соединены кабелем. Конкуренты решили заняться техническим шпионажем и перехватить трафик сети. Однако доступ они могут получить только к коммуникациям, расположенным в коридоре.

Смогут ли конкуренты прослушивать весь трафик, циркулирующий по сети?

Варианты ответов:

- а) да, смогут прослушивать весь трафик
- б) да, только трафик между четными компьютерами
- в) да, только трафик между нечетными компьютерами
- г) да, только трафик между четными и нечетными компьютерами
- д) нет

Ответ: а).

Условия задач заключительного этапа

Задача 1. Пин-код

У Альберта в качестве пин-кода от банковской карты используется такое простое четырехзначное число, для которого сумма первых трёх цифр равна последней цифре. После 10 неправильных попыток ввода пин-кода карта блокируется. Можно ли гарантированно ввести правильный пароль и не заблокировать карту, если Альберт его забыл? Ответ обоснуйте.

Решение

Простое четырехзначное число лежит в диапазоне от 1000 до 9999. Именно в этом диапазоне необходимо найти простые числа, удовлетворяющие условию "сумма первых трёх цифр равна последней цифре". Если представить простое число в виде $abcd$, то условие будет выглядеть следующим образом:

$$a+b+c=d$$

Для ускорения процесса поиска таких чисел можно написать программу.

Пример функции проверки простого числа на языке С.

```
// ФУНКЦИЯ проверки, является ли число простым
// INPUT - NUM - целое число
// RETURN - TRUE - число простое
//          FALSE - число непростое
bool is_prost(int num)
{
    int d = 2;                // делитель числа
    // цикл по делителю от 2-х до num/2
    while (d < num / 2)
    {
        // если остаток от деления num на d
        // нулевой, число - непростое, return false
        if (num % d == 0)
            return false;
        d++;                // увеличиваем делитель на 1
    }
    return true;
}
```

Пример функции проверки условия $a+b+c=d$ на языке С.

```
// ФУНКЦИЯ проверки условия a+b+c=d
// INPUT - NUM - целое число
// RETURN - TRUE - число удовлетворяет условию
//          FALSE - число НЕ удовлетворяет условию
bool check(int num)
{
    int a = num / 1000;        // 1-й разряд числа NUM
    int b = (num / 100) % 10; // 2-й разряд числа
                                NUM
    int c = (num / 10) % 10; // 3-й разряд числа NUM
    int d = num % 10;         // 4-й разряд числа NUM
```

```

    if (a + b + c == d)
        return true;
    return false;
}

```

Пример функции `main()`, выводящей на экран простые числа, удовлетворяющие условию.

```

int main()
{
    // Перебор всех 4-значных чисел от 1000 до 9999
    // и вывод на экран только простых и
    // удовлетворяющих условию check()
    for (int n = 1000; n < 10000; n++)
        if (is_prost(n) && check(n))
            printf("%d\n", n);
    return 0;
}

```

Результат выполнения программы выводит следующие числа:

```

1427
1607
2237
2417
3137
3407
4127
4217

```

Всего простых чисел, подходящих под условие – 8. Значит, если Альберт забыл пин-код, он сможет перебрать все возможные варианты без блокирования карты.

Ответ: да, подходящих под условия пин-кодов всего 8.

Задача 2. Тайное послание

В феврале 1890 г. начальник сыскной полиции г. Санкт-Петербурга Иван Дмитриевич Путилин проводил расследование очередного преступления. Имелся подозреваемый, но не хватало доказательств. При обыске в номере гостиницы, где проживал предполагаемый преступник, Иван Дмитриевич обнаружил на столе три предмета: листок бумаги с нарисованной от руки таблицей, книгу и перевёрнутую вверх дном кружку с надписью «13» на доньшке.

После недолгих размышлений Ивану Дмитриевичу удалось получить необходимую информацию для раскрытия преступления. Удастся ли это Вам? Ниже представлена таблица, нарисованная на найденном листе бумаги. Книга расположена в отдельном файле.

10, 4, 2	12, 2, 96	7, 14, 95	9, 2, 65						
		6, 8, 1	42, 11, 86						
				28, 11, FF					
					20, 5, 4				
				12, 7, 14	8, 4, 3				

Электронные материалы: электронная книга Стивенсон Роберт Луис - *Остров сокровищ* (96 стр.)
(http://v-olymp.ru/olmp_it/docs/2018/tasks/9-10/2/Стивенсон_Роберт_Луис_-_Остров_сокровищ.pdf).

Решение

Таблица состоит из 98 ячеек, столько же страниц в книге. Нетрудно догадаться, что каждая ячейка соответствует странице в книге.

Можно предположить, что первое число указывает на номер строки сверху, а второе - на номер слова в строке слева.

Третье число в ячейках не повторяется и лежит в диапазоне от 1 до 98, кроме значения FF. Можно предположить, что третье число указывает на следующую ячейку, а ячейка FF – это последняя ячейка.

Остается вопрос: с какой ячейки/страницы начинать?

В задании указано число «13», значит первое слово на 13-й странице, информация об этом слове содержится в ячейке 13: 6-я строка, 8-е слово – «ЗОЛОТО».

Следующее слово на странице 1: 10-я строка, 4-е слово – «ЛЕЖИТ».

Следующая страница 2 и т.д. (см. таблицу).

<i>Страница/ ячейка</i>	<i>№ строки</i>	<i>№ слова</i>	<i>Слово</i>
13	6	8	ЗОЛОТО
1	10	4	ЛЕЖИТ
2	12	2	В
96	8	4	СУНДУКЕ
3	7	14	В
65	28	11	КАЮТЕ
14	39	11	ТРУП
86	20	5	СПРЯТАН
4	9	2	В
95	12	7	ПЕЩЕРЕ

Ответ: Золото лежит в сундуке в каюте. Труп спрятан в пещере.

Задача 3. Скрытое сообщение

Взаимодействие между агентами осуществляется по каналу связи, позволяющему последовательно передавать несколько текстовых сообщений произвольного ненулевого размера, кратного 1 Кб. В ходе осуществления очередного сеанса связи было отправлено 3 сообщения общим размером 16 Кб. Для однозначного определения размеров и места каждого сообщения в сеансе связи агенту на приемной стороне потребовалось дополнительно сообщить значение, равное произведению их размеров, и тот факт, что сообщение наибольшего размера было отправлено первым. Помогите агенту определить размеры полученных сообщений с учетом порядка их отправки.

Решение

Рассмотрев все возможные комбинации длин сообщений (в сумме которые дают 16), можно заметить, что в большинстве из них нельзя однозначно определить последовательность сообщений, даже обладая информацией о том, что максимальной длины сообщение идет первым. Полученной информации будет достаточно только в том случае, когда два других сообщения будут обладать одинаковой длиной. Под такие критерии подпадают следующие наборы:

$$150=6*5*5$$

$$128=8*4*4$$

$$58=12*2*2$$

$$14=14*1*1$$

$$90=10*3*3$$

Ответ: 6*5*5 или 8*4*4 или 12*2*2 или 14*1*1 или 10, 3, 3.

Задача 4. ZIP

В ходе исследования компьютера одного из сотрудников организации были обнаружены подозрительные файлы. Предварительный анализ показал, что среди них есть файлы, в которых сотрудник хранил пароли к данным в определенном формате:

{XXXX-XXXX-XXXX-XXXX}

Непроанализированным остался единственный файл HowTo.docx. Помогите найти в этом файле возможный пароль.

Электронные материалы: файл HowTo.docx

(http://v-olymp.ru/olmp_it/docs/2018/tasks/9-10/4/HowTo.docx),

файл Proc.zip (http://v-olymp.ru/olmp_it/docs/2018/tasks/9-10/4/Proc.zip).

Решение

Можно обратить внимание, что первые байты файла HowTo.docx совпадают с первыми байтами файла Proc.zip. Это означает, что файл формата .DOCX представляет собой zip-архив.

```

00000000: 50 4B 03 04 14 00 06 00|08 00 00 00 21 00 E7 21 |PK.....!.z!
00000010: 07 5D 70 01 00 00 07 05|00 00 13 00 08 02 5B 43 |.]p...4.....[C
00000020: 6F 6E 74 65 6E 74 5F 54|79 70 65 73 5D 2E 78 6D |ontent_Types].xm
00000030: 6C 20 A2 04 02 28 A0 00|02 00 00 00 00 00 00 00 |l ŷ. ( . ....
00000040: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000050: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000060: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000070: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000080: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....

```

Рисунок 1 – Содержимое файла HowTo.docx

```

00000000: 50 4B 03 04 14 00 02 00|08 00 07 70 52 4D 83 DF |PK.... .PRMгЯ
00000010: 2A D4 8B 00 00 00 00 00|00 00 08 00 00 00 50 72 |*Ф<...#.....Pr
00000020: 6F 63 2E 74 78 74 3D 8E|41 0E 82 30 10 45 AF F2 |oc.txt=ќА.,0.EЇт
00000030: C3 1A 3D 86 6B 17 9E DA|9D 37 18 4B 47 4A A1 53 |Г.=ќk.ћьк7.KGJŸS
00000040: 66 1A 43 08 48 EC 46 97|2F 79 3F EF DF A5 F3 04 |f.C.HMF-/y?няггy.
00000050: B7 41 2C 8E C7 42 6C AC|7B 0B 13 E8 24 AF 4A E8 |-A,ќ3B1-ζ...и$ЇЖи
00000060: 29 CB 94 82 ED 30 8E 65|65 86 0A FB 90 38 05 AE |)л",н0ћееќ.ыћ8.0
00000070: 22 45 2C D5 96 24 78 16|CF 06 2F 6A 25 57 74 F0 |"E,x-Ѕx.П./j%Wtp
00000080: 81 46 E9 18 CD E3 BF 80|E2 72 BB 9E 94 67 C7 B3 |ѓFй.HгїЂвр>ћ"q3i

```

Рисунок 2 – Содержимое файла Proc.zip

Переименовав файл HowTo.docx в HowTo.zip можно его открыть любой программой-архиватором (WinZip или WinRar). В результате отобразится содержимое архива в виде некоторого количества файлов и папок.

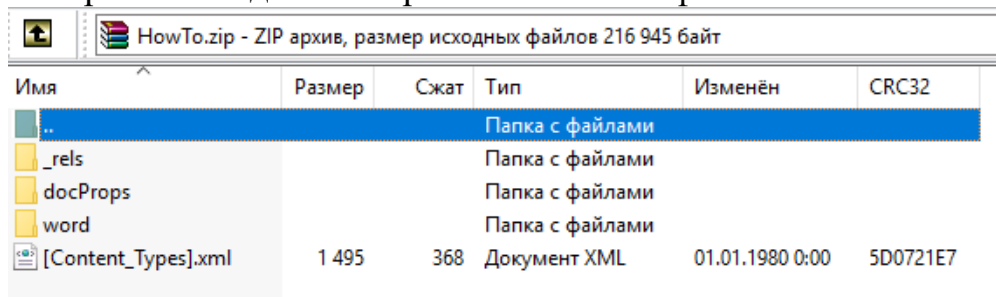
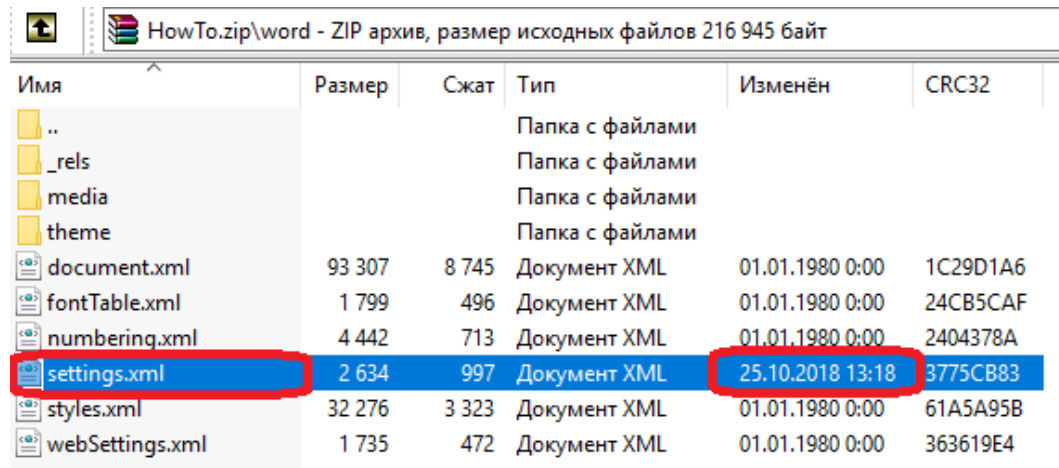


Рисунок 3 – Содержимое файла HowTo.zip

Выполнив поиск данных, подходящих под формат пароля, по всем файлам, содержащимся в архиве, можно определить, что пароль спрятан в файле *settings.xml*.

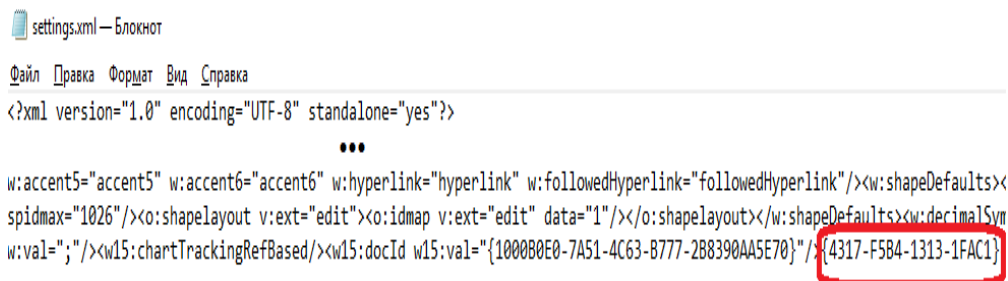
Это можно увидеть одним из способов:

- поиск в содержимом файлов текста формата {XXXX-XXXX-XXXX-XXXXXX},
- анализируя дату изменения файла.



Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Папка с файлами		
_rels			Папка с файлами		
media			Папка с файлами		
theme			Папка с файлами		
document.xml	93 307	8 745	Документ XML	01.01.1980 0:00	1C29D1A6
fontTable.xml	1 799	496	Документ XML	01.01.1980 0:00	24CB5CAF
numbering.xml	4 442	713	Документ XML	01.01.1980 0:00	2404378A
settings.xml	2 634	997	Документ XML	25.10.2018 13:18	3775CB83
styles.xml	32 276	3 323	Документ XML	01.01.1980 0:00	61A5A95B
webSettings.xml	1 735	472	Документ XML	01.01.1980 0:00	363619E4

Рисунок 4 – Измененный файл



```

settings.xml – Блокнот
Файл  Правка  Формат  Вид  Справка
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
...
w:accent5="accent5" w:accent6="accent6" w:hyperlink="hyperlink" w:followedHyperlink="followedHyperlink"/><w:shapeDefaults>
spidmax="1026"/><o:shapelayout v:ext="edit"><o:idmap v:ext="edit" data="1"/></o:shapelayout></w:shapeDefaults><w:decimalSym
w:val=""/><w15:chartTrackingRefBased/><w15:docId w15:val="{1000B0E0-7A51-4C63-B777-2B8390AA5E70}"/>{4317-F5B4-1313-1FAC1}

```

Рисунок 5 – Содержимое файла settings.xml

Ответ: {4317-F5B4-1313-1FAC1}

Задача 5. Подбор пароля

В системе безопасности используется n -разрядный ключ ($0 < n \leq 64$). Для проверки вводимого пользователем ключа вызывается функция *Check()*, которой он передается в качестве параметра. Результатом проверки является число совпадающих разрядов между введенным пользователем и эталонным ключами. Введенный ключ считается правильным, если функция *Check()* вернула значение, равное n . Предложите алгоритм, который гарантированно угадывает значение ключа не более чем за n вызовов функции проверки.

Листинг функции Check()

```

const __int64 KEY;           // эталонный ключ

/* ВХОДНЫЕ ПАРАМЕТРЫ:
* key      - значение проверяемого ключа

```

```

*           (последний разряд - самый младший)
* n         - количество разрядов ключа
* СТОРОННИЕ ПЕРЕМЕННЫЕ:
* KEY      - эталонное значение ключа (объявлено вне функции)
* ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ:
* int      - количество верных разрядов ключа key
*/
int Check(__int64 key, int n)
{
    int res = 0;
    for (int i = 0; i < n; i++)
    {
        if ( (key & (0x00000000000000001 << i)) ==
            (KEY & (0x00000000000000001 << i)) )
            res++;
    }
    return res;
}

```

Решение

Как работает функция Check():

- 1) В цикле от 0 до N (длина ключа в битах) происходит сравнение i-го бита проверяемого ключа с эталонным, начиная с крайнего правого.
- 2) Если i-й бит проверяемого ключа совпадает с i-м битом эталонного ключа, счетчик res увеличивается на 1.
- 3) По окончании цикла счетчик содержит число совпавших бит у проверяемого и эталонного ключа. Если счетчик равен N – ключи равны (все биты одинаковые).

По условиям задачи необходимо разработать алгоритм определения эталонного ключа не более, чем за N запусков функции Check().

Предлагается следующее решение:

- 1) Изначально принять ключ, равный 00000...0. Проверяемый бит $k = 0$ (первый).
- 2) Подать его на вход функции Check(). Запомнить результат (res1).
- 3) Инвертировать значение k-го бита (поменять с «0» на «1» или наоборот).
- 4) Вызвать функцию Check() с измененным ключом, поданным на вход. Сравнить результат выполнения функции (res2) с результатом, полученным в п.2 (res1). Если $res2 > res1$, оставить ключ без изменения. Если $res2 < res1$ инвертировать обратно измененный k-й бит.
- 5) Увеличить значение k на 1 ($k = k + 1$).
- 6) Если $k = N - 1$ (последний разряд), то проанализировать результат функции Check() (res1). Если $res1 = N$ – ключ найден. Если $res1 = N - 1$, необходимо проинвертировать k-й бит – ключ найден. Конец.
- 7) Переход на п.2.

Примечание. В данном алгоритме абсолютно неважно, какое начальное значение ключа будет использовано, и в каком порядке будут проверяться биты (слева-направо или справа-налево). Важно, что на каждом шаге изменяется ТОЛЬКО один бит.

Ответ: Начиная с любого набора изменять по одному биту. Если результат выполнения функции Check() изменился в БОльшую сторону – угадали значение изменённого бита. Если результат изменился в мЕньшую сторону – НЕ угадали значение измененного бита, необходимо вернуть исходное значение этого бита.

11 КЛАСС

Условия задач отборочного этапа

Задача 1

Политика безопасности ОС запрещает задавать для учетных записей пользователей пароли (pwd), совпадающие с именами (login) этих учетных записей, т.е. не разрешается равенство $\text{pwd} = \text{login}$.

При регистрации нового пользователя в системе вычисляется хеш-значение пароля с помощью функции GetHash, которое и сохраняется в ОС.

Кроме того, администратор периодически выполняет проверку зарегистрированных пользователей и блокирует учетные записи, хеш-значения от имени которых совпадают с хеш-значениями от их паролей.

Укажите, какое минимальное количество первых символов имени и пароля учетной записи должны совпадать (т.е. начала строк одинаковы, а оставшиеся части должны различаться), чтобы администратор заблокировал учетную запись.

Синт	<pre> char *GetHash(const char* inStr) /* Входные параметры: inStr - строка пароля Возвращаемое значение: строка хэша пароля */ { const char Symbols[] = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890"; unsigned SymbolsLength = strlen(Symbols); unsigned inStrLen = strlen(inStr); const unsigned BufferLength = 20; static char Buffer[BufferLength + 1]; Buffer[BufferLength] = '\0'; const unsigned HashLength = 8; static char Hash[HashLength + 1]; Hash[HashLength] = '\0'; unsigned int BufStrLen = (inStrLen > BufferLength) ? BufferLength : inStrLen; int i = 0; for (; i < BufStrLen; i++) Buffer[i] = inStr[i]; for (; i < BufferLength; i++) Buffer[i] = '0' + (i - BufStrLen); const int p = 19; unsigned short int hash = 0, p_pow = 1; for (i = 0; i < BufferLength && i < HashLength*2; i += 2) { hash += (Buffer[i] - '0' + 1) * p_pow; p_pow *= p; hash += (Buffer[i + 1] - '0' + 1) * p_pow; p_pow *= p; Hash[i >> 1] = Symbols[hash % SymbolsLength]; } return Hash; } </pre>
------	--

Ответ: 16.

Задача 2

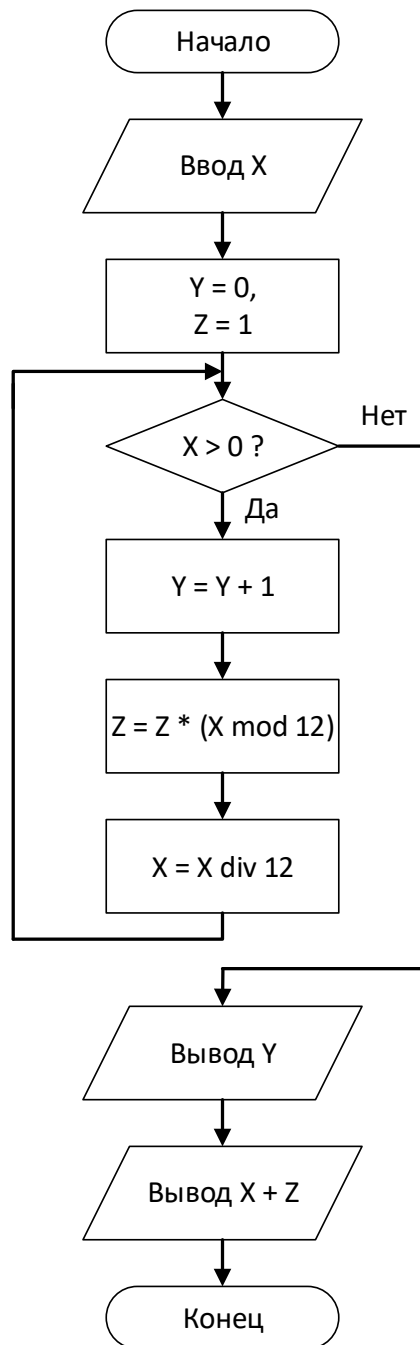
На рисунке представлена блок-схема алгоритма обработки целого числа X , вводимого пользователем.

Используемые в блок-схеме операции:

div – целая часть от деления;

mod – остаток от деления.

Какое максимальное число необходимо ввести, чтобы было выдано 3, а затем 48?



Ответ: 1225

Задача 3

Для поддержания работоспособности компьютерной системы, отвечающей за аутентификацию пользователей, реализована RAID-система из двух зеркальных томов (записи на таких томах идентичны, т.е. "зеркалированы"). Из-за требований по энергопотреблению тома реализованы физически как твердотельные устройства памяти. Ресурс каждого устройства – $4 \cdot 10^6$ операций чтения/записи. Из соображений надежности система собирается из двух устройств, но еще используется третье устройство, которое периодически заменяет одно из работающих устройств так, что износ всех трех устройств остается равномерным.

Сколько такая система сможет проработать без сбоев по операции чтения/записи, если интенсивность запросов на аутентификацию составляет 500 запросов в час и эта интенсивность постоянна за время работы?

Варианты ответов:

- a) От 6 до 12 месяцев
- b) Менее 6 месяцев
- c) Более 2 лет
- d) От 1 года до 2 лет

Ответ: d).

Задача 4

Машинный код команды безусловной передачи управления – JMP – для 32-разрядных процессоров семейства Intel x86 имеет вид E9 xx xx xx xx, где каждая пара xx - это шестнадцатеричные цифры байта, входящего в целевой адрес передачи управления) Т.е. код вида E9 12 34 56 78 передает управление на адрес $\Delta + 78\ 56\ 34\ 12$ (в шестнадцатеричном виде), где Δ – адрес размещения команды, следующей за командой JMP.

Вирус использует технологию, при которой команда безусловной передачи управления на адрес начала тела вируса заменяет некоторую соразмерную команду программы, дописываемую в конец тела вируса. После выполнения тела вируса и дописанной команды осуществляется передача управления на оригинальный код.

На сколько байтов минимально увеличится размер заражаемого файла, если изначально тело вируса имело размер 10 байтов, а программа после заражения сохраняет свою функциональность полностью?

Ответ: 20.

Задача 5

В исполняемый файл PROG.EXE злоумышленник внедрил секретное текстовое сообщение, не нарушив при этом функциональность программы.

Известно, что сообщение представлено тремя отдельными фрагментами, для определения границ которых используется единая однобайтовая метка.

Какое сообщение было внедрено в файл? В ответе укажите только само сообщение с учетом регистра и пробелов.

Ответ: Application and Service Parameters

("Application a" + "nd Service Para" + "meters" - mask - 0xE7 (5 раз))

Условия задач заключительного этапа

Задача 1. Шифр

Сотруднику для анализа был предоставлен перехваченный фрагмент зашифрованного текста. Известно, что некоторые предложения исходного текста начинаются с фразы «устроить атаку» (регистр не учитывается). Каждый символ исходного текста закодирован тремя цифрами, а точка имеет код «200», после точки пробел не ставится.

```
020012021102101002021112000111002211212102012002210210200002022120
002011001212100212200220221211210101212111010020111001112212121202
111222011022010221200201002120111222201011011102202102101001022221
220112202201220210220112200020012021102101002021112000111002211212
102012002210122122101121010012010011201012010200020012021102101002
02111200011100221121210201200221022021212202201222221101220011200
100020200220022121210100021210202102100212221200210102000200011211
110212121010022211022121012212000122100120122110001001120200001201
022012022021112122100111100002102122002000221010100211222110102000
222120020202200012210001111100110021001100202021200222011112121011
200211100200111221200000111210201220221211020111220121102110220122
010000112111222221020022112021020202120200020012021102101002021112
000202021202121020202212200101100121201120202221200020012021102101
002021112000111002211212102012002210212120010100100211222120111021
002022012012002100001200020012021102101002021112000202021202121020
212010202212210100121100120022012200020012021102101002021112000111
002211212102012002210201210020020221102211011121202101121201222010
200020012021102101002021112000202021202121020200122001122101112200
222002000200201100121022211221010001122211121001201112201020011012
212100111201111212122200020012021102101002021112000111002211212102
01200221011222201202220112022220210212200020012021102101002021112
000202021202121020022001101102122220111021100001001202020211220211
200020012021102101002021112000202021202121020001120010011201110000
010012011220221200020012021102101002021112000202021202121020201012
210201011010110002102221200210102021210122122012012201212022210021
201012022112022100022111112020000122200212121101201211022200111022
210222001201202110111021210100122120121122112002200011210000212121
111112122212012022211111120121210002211200122212221001022002120212
000001222002020000220121101211210102020212201121200010112121211020
021021001221211111222010020211220211212102200020012021102101002021
112000111002211212102012002210022100120022110002010212211212120120
022111022222010200
```

Определите, как будет записано слово «СКОРОСТЬ» с использованием представленного шифра.

Решение:

Первым шагом необходимо разбить фрагмент на предложения (код точки – «200»). Необходимо учитывать, что длина предложения должна быть кратна трем. Для удобства разобьем весь фрагмент на группы по 3 цифры (символы).

020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
 210 210 **200**
 002 022 120 002 011 001 212 100 212 **200**
 220 221 211 210 101 212 111 010 020 111 001 112 212 121 202 111
 222 011 022 010 221 **200**
 201 002 120 111 222 201 011 011 102 202 102 101 001 022 221 220
 112 202 201 220 210 220 112 **200**
 020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
 210 122 122 101 121 010 012 010 011 201 012 010 **200**
 020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
 210 220 212 122 022 012 222 221 101 220 011 **200**
 100 020 **200**
 220 022 121 210 100 021 210 202 102 100 212 221 **200**
 210 102 000 **200**
 011 211 110 212 121 010 022 211 022 121 012 212 000 122 100 120
 122 110 001 001 120 **200**
 001 201 022 012 022 021 112 122 100 111 100 002 102 122 002 000
 221 010 100 211 222 110 102 000 222 120 020 202 **200**
 012 210 001 111 100 110 021 001 100 202 021 **200**
 222 011 112 121 011 **200**
 211 100 **200**
 111 221 **200** 000 111 210 201 220 221 211 020 111 220 121 102 110
 220 122 010 000 112 111 222 221 020 022 112 021 020 202 120 **200**
 020 012 021 102 101 002 021 112 000 202 021 202 121 020 202 212
200
 101 100 121 201 120 202 221 **200**
 020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
 210 212 120 010 100 100 211 222 120 111 021 002 022 012 012 002
 100 001 **200**
 020 012 021 102 101 002 021 112 000 202 021 202 121 020 212 010
 202 212 210 100 121 100 120 022 012 **200**
 020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
 210 201 210 020 020 221 102 211 011 121 202 101 121 201 222 010
200
 020 012 021 102 101 002 021 112 000 202 021 202 121 020 **200**
 122 001 122 101 112 **200**
 222 002 000 **200**
 201 100 121 022 211 221 010 001 122 211 121 001 201 112 201 020
 011 012 212 100 111 201 111 212 122 **200**

...

В условии сказано, что фраза «УСТРОИТЬ АТАКУ» встречается больше одного раза. Эта фраза состоит из 14-ти символов (включая пробел) и должна быть закодирована 42-мя цифрами (по 3 на символ). Необходимо найти повторяющиеся фрагменты в началах предложений, длиной 14 символов (42 цифры). Такие повторяющиеся фрагменты есть.

020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
210 210 200
 002 022 120 002 011 001 212 100 212 **200**
 220 221 211 210 101 212 111 010 020 111 001 112 212 121 202 111
 222 011 022 010 221 **200**
 201 002 120 111 222 201 011 011 102 202 102 101 001 022 221 220
 112 202 201 220 210 220 112 **200**
020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
210 122 122 101 121 010 012 010 011 201 012 010 200

020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
210 220 212 122 022 012 222 221 101 220 011 **200**
 100 020 **200**
 220 022 121 210 100 021 210 202 102 100 212 221 **200**
 210 102 000 **200**
 011 211 110 212 121 010 022 211 022 121 012 212 000 122 100 120
 122 110 001 001 120 **200**
 001 201 022 012 022 021 112 122 100 111 100 002 102 122 002 000
 221 010 100 211 222 110 102 000 222 120 020 202 **200**
 012 210 001 111 100 110 021 001 100 202 021 **200**
 222 011 112 121 011 **200**
 211 100 **200**
 111 221 **200** 000 111 210 201 220 221 211 020 111 220 121 102 110
 220 122 010 000 112 111 222 221 020 022 112 021 020 202 120 **200**
020 012 021 102 101 002 021 112 000 202 021 202 121 020 202 212
200
 101 100 121 201 120 202 221 **200**
020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
210 212 120 010 100 100 211 222 120 111 021 002 022 012 012 002
 100 001 **200**
020 012 021 102 101 002 021 112 000 202 021 202 121 020 212 010
 202 212 210 100 121 100 120 022 012 **200**
020 012 021 102 101 002 021 112 000 111 002 211 212 102 012 002
210 201 210 020 020 221 102 211 011 121 202 101 121 201 222 010
200
020 012 021 102 101 002 021 112 000 202 021 202 121 020 200
 122 001 122 101 112 **200**
 222 002 000 **200**
 201 100 121 022 211 221 010 001 122 211 121 001 201 112 201 020
 011 012 212 100 111 201 111 212 122 **200**
 ...

Найдено 2 повторяющихся фрагмента:

020 012 021 102 101 002 021 112 000 111 002 211 212 102 012
002 210

И

020 012 021 102 101 002 021 112 000 202 021 202 121 020

По длине фрагмента и коду можно определить, что во втором фрагменте закодирована фраза «УСТРОИТЬ АТАКУ». Из него можно извлечь коды для букв, входящих в эту фразу:

Символ	Код
У	020
С	012
Т	021
Р	102
О	101
И	002
Ь	112
ПРОБЕЛ	000

А	202
К	121

Слово «СКОРОСТЬ» состоит из букв, для которых известен код, поэтому ответ написать не составит труда.

Ответ: 012 121 101 102 101 012 021 112

Задача 2. Удаленный файл

Для организации файловой системы на сервере компании используется структура хранения данных, включающая в себя три последовательно расположенных основных секции:

Таблица 1 (288 байт)	Таблица 2 (510 байтов)	Данные
-------------------------	---------------------------	--------

Каждый файл разбивается на блоки размером не более 10 байт, информация о которых хранится в секции «Таблица 2», содержащей N ячеек и имеющей следующую структуру:

Ячейка с индексом 1			Ячейка с индексом N	
Адрес блока данных (1 байт)	Индекс следующей ячейки (1 байт)	...	Адрес блока данных (1 байт)	Индекс следующей ячейки (1 байт)

где «Адрес блока данных» – смещение блока данных файла в секции «Данные» относительно ее начала; «Индекс следующей ячейки» – порядковый номер ячейки (в секции «Таблица 2»), содержащей информацию о следующем блоке данных файла, либо 0x00, если достигнут конец файла.

Для задания начала каждого файла используется секция «Таблица 1», содержащая M ячеек и имеющая следующую структуру:

Ячейка с индексом 1			Ячейка с индексом M	
Имя файла (8 байт)	Индекс первой ячейки (1 байт)	...	Имя файла (8 байт)	Индекс первой ячейки (1 байта)

где «Имя файла» – имя файла; «Индекс первой ячейки» – индекс ячейки (в секции «Таблица 2»), содержащей информацию о первом блоке данных файла.

Все неиспользуемые ячейки секции «Таблица 1» заполнены байтами 0x00. Для повышения скорости работы с файловой системой при удалении файла обнуляются (заполняются байтами 0x00) только соответствующие ему ячейки секции «Таблица 1».

В результате ошибки администратора с сервера был удален один текстовый файл. Предложите алгоритм восстановления и приведите содержимое удаленного файла, имея в распоряжении файл *DiskImage.bin*, являющийся полной копией содержания и структуры файловой системы и данных, находящихся на диске, с которого был удален файл.

К задаче прилагается: файл *DiskImage.bin*.

Решение:

Исходя из структуры таблиц, для решения задачи необходимо найти все ячейки в «Таблице 2», ссылок на которые нет из других ячеек «Таблицы 2» (продолжение файла) и из «Таблицы 1» (начало файла). Для упрощения задачи поиска таких ячеек необходимо написать программу автоматизирующую соответствующие операции. Так как был удален только 1 файл, то такая ячейка будет единственной. В дальнейшем необходимо определить все блоки данных, которые соответствуют файлу путем считывания индексов всех ячеек до момента, пока не будет достигнут конец файла (значение индекса следующей ячейки будет равно 0x00).

Ответ: Был удален файл *ls.txt*

Задача 3. Скрытое сообщение

Алексей получил электронное почтовое сообщение со следующим содержимым: «Отправляю тебе мое кодовое слово. С уважением, Сергей». К сообщению были прикреплены файлы *FirstFile.bmp*, *SecondFile.bmp*. Помогите Алексею определить полученное от Сергея кодовое слово.

Электронные материалы: файлы *FirstFile.bmp*, *SecondFile.bmp*.
(http://v-olymp.ru/olmp_it/docs/2018/tasks/11/3/files.rar).

Решение

Визуально файлы не различимы. Стоит проанализировать содержимое файлов в бинарном формате.

Сравнив содержимое файлов, можно увидеть, что отличаются 64 байта:

00C0: F7 F5 F5 FC FA FA FF FD чххъъъъъ	00C0: F7 F5 F5 FC FA FA FF FD чххъъъъъ
00C8: FD FF FE FE FE FF FF FF зяююяяяя	00C8: FD FF FE FE FE FE FF FF зяююяяяя
00D0: FF FF FF FD FD FC FA FA яяяээээъ	00D0: FF FF FF FD FD FC FA FA яяяээээъ
00D8: FB F9 F9 FF FF FF FF FF ъщяяяяяя	00D8: FB F9 F9 FF FF FF FF FF ъщяяяяяя
00E0: FF FC FA FA F7 F5 F5 FD яъъъчххэ	00E0: FF FC FA FA F7 F5 F5 FD яъъъчххэ
00E8: FB FB FF FF FF FF FE FE ъыяяяяюю	00E8: FB FB FF FF FF FF FE FE ъыяяяяюю
00F0: FC FC FC FC FC FC FC FC ъъъъъъъъ	00F0: FC FC FC FC FC FC FC FC ъъъъъъъъ
00F8: FC FC FC FC FC FC FC FC ъъъъъъъъ	00F8: FC FC FC FC FC FC FC FC ъъъъъъъъ
0100: FC FC FC FC FC FC FC FC ъъъъъъъъ	0100: FC FD FC FC FC FC FC FC ъъъъъъъъ
0108: FC FC FC FC FC FC FC FC ъъъъъъъъ	0108: FC FC FC FC FC FC FC FC ъъъъъъъъ
0110: FC FC FC FC FC FC FC FC ъъъъъъъъ	0110: FC FC FC FC FC FC FC FC ъъъъъъъъ

Рисунок 1 – Различия в файлах

Байты отличаются только последним разрядом: $FF \rightarrow FE$, $FC \rightarrow FD$. В первом байте последний бит поменяли с «1» на «0», во втором «0» \rightarrow «1».

Можно сделать предположение, что измененный бит содержит информацию, каждый измененный байт несёт 1 бит информации. Учитывая, что измененных байтов 64, можно сделать вывод, что сообщение состоит из 8-ми байт.

Собирая измененные байты по порядку их появления от начала файла и извлекая модифицированные биты, можно собрать 8 байт информации.

Логично предположить, что информация является сообщением, и каждый байт представляет собой символ в ASCII-формате, то есть значение байта - это код символа в ASCII-таблице.

Получив значение 8-ми байт и сверившись с ASCII-таблицей можно получить ответ.

Ответ: Security

Задача 4. Гамма

Текстовое сообщение было зашифровано методом «двоичного гаммирования», т.е. путем выполнения операции «побитового исключающего ИЛИ» между байтами исходного сообщения и ключа длиной 2 байта. После применения операции «побитового исключающего ИЛИ» к байтам ключа и первым двум байтам сообщения, ключ сдвигается на 1 бит вправо относительно исходного текста, и операция выполняется повторно. Результат выполнения операции «побитового исключающего ИЛИ» сохраняется в соответствующих разрядах зашифрованного текста. Шифрование заканчивается, когда операция применяется к двум последним байтам сообщения.

```
35 17 1E 1A 0F 1E 1A 18 11 10 1F 0E 12 11 0E 0D 03 DF 10
0F 1A 1D 04 07 1A DF 1D 0E 1A 70 43
```

Определите исходное сообщение и значение ключа, если известно, что сообщение может содержать только буквы, цифры, пробелы и знаки препинания.

Решение:

Первым делом необходимо понять схему шифрования: каждый раз 2-байтный ключ сдвигается относительно текста на 1 разряд вправо. Это означает, что начиная с 3-го байта текста, все остальные, кроме последних двух байтов, складывается по модулю 2 со всеми разрядами ключа. То есть каждый разряд байта текста последовательно складывается со всеми разрядами ключа. Это означает, что все разряды текста складываются с одним и тем же значением, равным сумме по модулю 2 всех разрядов ключа.

Если сумма по модулю 2 всех разрядов ключа равна 0 – результат сложения не изменится. Если сумма по модулю 2 всех разрядов ключа равна 1 – результат сложения будет инверсией исходного текста.

Если попытаться декодировать сообщение по ASCII-таблице, смысловой текст не получится. Значит необходимо инвертировать разряды зашифрованного текста:

```
CA E8 E1 E5 F0 E1 E5 E7 EE EF E0 F1 ED EE F1 F2 FC 20 EF
F0 E5 E2 FB F8 E5 20 E2 F1 E5 8F BC
```

Инвертированный фрагмент можно преобразовать в текст с использованием ASCII-таблицы:

«Кибербезопасность превыше всеЦj»

Как видно, последние 2 байта не являются буквами. Это так, поскольку последние 2 байта складывались с ключом ровно 1 раз, причем каждый разряд текста складывался по модулю 2 с соответствующим разрядом ключа.

Из смыслового содержания сообщения можно предположить, что последние 2 байта – это буквы «g» и «o».

При известном исходном тексте и известном зашифрованном тексте (70 43) не составляет труда найти ключ. Для этого необходимо сложить по модулю 2 исходный текст (ASCII-код букв «g» и «o») и зашифрованный текст:

```
E3 EE
 70 43
-----
93 AD
```

Ответ: сообщение «Кибербезопасность превыше всего», ключ шифрования – 93 AD.

Задача 5. Ключи

Взаимодействие между агентами осуществляется по каналу связи, позволяющему последовательно передавать несколько ключей шифрования произвольной длины. Длина каждого ключа кратна байту. В ходе осуществления очередного сеанса связи было отправлено 3 ключа шифрования:

```
44 41 54 4B 42 4A 41 43 4C 41 50 4C 56 58 4B 46 56 4C 41
50 41 54 4A 4C 44 4A
```

Для возможности однозначного определения длин всех ключей шифрования агенту на приемной стороне дополнительно (по некоторому другому каналу связи) сообщили значение, равное произведению их длин, и тот факт, что ключ наибольшей длины содержит последовательность байтов 0x41544A. Помогите агенту восстановить каждый из трех полученных ключей.

Решение:

Исходя из того, что предоставленных данных о сумме (26) и произведении размеров пакетов недостаточно, можно сделать вывод – существует несколько наборов значений, соответствующих размерам сообщений, имеющих одинаковую сумму и произведение. Под такие критерии подпадают следующие наборы:

$$288 = 18 * 4 * 4 = 12 * 12 * 2$$

$$270 = 18 * 5 * 3 = 15 * 9 * 2$$

$$126 = 21 * 3 * 2 = 18 * 7 * 1$$

Наличие информации о содержимом самого большого сообщения позволяет однозначно определить размеры переданных сообщений только для случая, когда произведение размеров равно 288, где встречаются 2 сообщения имеющих одинаковую длину. Для других ситуаций данная информация не устраняет неопределенность (для 270 – «самое короткое сообщение имеет длину 3 или 2?», и то и другое могут поместиться с 24 по 26 байты; для 126 - аналогично). Следовательно, с учетом информации о наличии сообщения, обладающего максимальной длиной, можно сделать вывод - размеры переданных сообщений равны 18, 4 и 4 байтов.

Далее необходимо определить, где начинается самое длинное сообщение. Так как 0x41544A начинается с 21 байта и заканчивается 23 байтом, то расположение сообщений в зависимости от их длины в потоке данных может быть только следующим: 4, 4, 18.

Ответ: Были переданы следующие ключи:

44 41 54 4B

42 4A 41 43

4C 41 50 4C 56 58 4B 46 56 4C 41 50 41 54 4A 4C 44 4A

Критерии оценивания заданий

Каждая задача заключительного этапа олимпиады при проверке работ оценивается по системе: «-», « $\bar{+}$ », « \pm », «+»:

«-» — отсутствует решение задачи;

« $\bar{+}$ » — отсутствует логически законченное решение задачи (решено менее половины задачи);

« \pm » — логически законченное решение с неточностями (ответ может быть неправильным);

«+» — полностью правильное решение задачи и правильный ответ.

Затем, с учётом сложности задач, осуществляется перевод результатов проверки в баллы. Для каждой параллели классов (8-10 и 11) разработана своя система баллов (см. Таблицы 1, 2) и критерии определения победителей и призеров (см. Таблицу 3), принятые на заседании жюри олимпиады.

Таблица 1

Критерии оценивания задач олимпиады для 8-10 классов

	1 задача	2 задача	3 задача	4 задача	5 задача
-	0	0	0	0	0
$\bar{+}$	1	2	2	2	2
\pm	2	3	4	3	3
+	3	4	5	4	4

Максимальное количество баллов за работу – 20.

Таблица 2

Критерии оценивания задач олимпиады для 11 класса

	1 задача	2 задача	3 задача	4 задача	5 задача
-	0	0	0	0	0
$\bar{+}$	1	1	2	2	2
\pm	3	2	3	4	5
+	4	4	5	5	6

Максимальное количество баллов за работу – 23.

Таблица 3

Критерии определения победителей и призеров олимпиады

Возрастная категория	1 место	2 место	3 место
11 класс	16 баллов и более	13-14 баллов	10-12 баллов
10 класс	15 баллов и более	12-13 баллов	9-11 баллов
9 класс	15 баллов и более	12-13 баллов	9-11 баллов
8 класс	15 баллов и более	12-13 баллов	9-11 баллов